INRIA

UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
BP 105
78153 Le Chesnay Cedex
France
Tél. (1) 39 63 55 11

Rapports de Recherche

N° 867

COMPLEXITY ANALYSIS OF TERM REWRITING SYSTEMS

Christine CHOPPY
Stéphane KAPLAN
Michèle SORIA

JUILLET 1988

COMPLEXITY ANALYSIS OF TERM REWRITING SYSTEMS

Christine Choppy, Stéphane Kaplan * and Michèle Soria ***

ABSTRACT. In this paper, we address the question of the quantitative evaluation of term rewriting systems. We assume these systems to be canonical, and correspond to algebraic specifications that define a family of so-called derived operators. Let f be a derived operator and $t_1, ..., t_p$ be terms in normal form, our measure of cost is the number of rewriting steps between $f(t_1, ..., t_p)$ and its normal form. We are interested in the average cost of f on a p–tuple of terms as a function of the total size (number of symbols) of the p–tuple:

$$\bar{C}_n = \sum \text{cost}(f(t_1, ..., t_p))/N_n \ ,$$

where the sum is taken over all p–tuples of size n, and N_n is the number of p–tuples of size n.

We systematically study the asymptotic evaluation of average costs via the introduction of formal series, using techniques developped for the complexity analysis of algorithms. Our results allow costs to be computed without any explicit manipulation of series: We provide the user with ready–to–use formulae that depend only on syntaxical characteristics of the rewriting system, and the amount of size computations is extremely reduced.

ANALYSE DE COMPLEXITE DES SYSTEMES DE RECRITURE

RÉSUMÉ. Dans cet article, nous nous intéressons à l'évaluation quantitative des systèmes de récriture. Nous supposons que ces systèmes sont canoniques et correspondent à des spécifications algébriques où sont définis des opérateurs dérivés. Etant donnés f un opérateur dérivé, et $t_1, ..., t_p$ des termes en forme normale, notre mesure de coût est le nombre de pas de récriture pour arriver à la forme normale. Nous nous intéressons au coût moyen de f sur un p–uplet de termes en fonction de la taille totale (en nombre de symboles) du p–uplet:

$$\bar{C}_n = \sum \text{coût}(f(t_1, ..., t_p))/N_n \ ,$$

où la somme est prise sur tous les p–uplets de taille n, et N_n est le nombre de p–uplets de taille n.

Nous étudions l'évaluation asymptotique des coûts moyens de manière systématique, à l'aide de séries formelles, en utilisant des techniques développées pour l'analyse de complexité des algorithmes. Nos résultats permettent d'obtenir une estimation des coûts sans avoir à manipuler explicitement ces séries: en effet les résultats sont exprimés en fonction de caractéristiques syntaxiques du système de récriture, et les calculs à effectuer sont extrêmement réduits.

---

* *LRI Université Paris-Sud 91405–Orsay*

** *LRI and INRIA, Rocquencourt 78153–Le Chesnay (France)*

COMPLEXITY ANALYSIS OF TERM REWRITING SYSTEMS

C. CHOPPY, S. KAPLAN, M. SORIA
Laboratoire de Recherche en Informatique
U.A. C.N.R.S. 410
Université Paris-Sud, Bât 490
F-91405 ORSAY Cédex, FRANCE
net : mcvax!inria!lri!cc, mcvax!inria!lri!kaplan, mcvax!inria!lri!soria

Introduction

Algebraic specifications are now widely used for data structuring and they turn out to be quite useful for various aspects of program development, such as prototyping, assisted program construction, proving properties, etc [BCV 85, FG 84, FGJM 84, GHW 85 or GH 86a and b, Kap 86]. Some of these applications require to add a notion of computation to algebraic specifications, for instance by providing a (convergent) rewrite rule system that expresses the properties of the operators. In this context, it may be of prime interest to define a notion of algorithmic complexity for an algebraic specification, or, more precisely, a notion of complexity for each operator defined in the specification. Computing operator complexity within a given specification helps understanding how evaluation costs are distributed ; it may single out "costly" operators, and motivate the search for an equivalent, but "cheaper", specification.

In [CLR 80], the cost of a term is defined as the number of rewriting steps for reducing it to its normal form, and the cost of an operator is defined as the general cost of a term obtained by applying this operator to terms in normal form. In this paper, we further formalize this notion of operator complexity and investigate its computation through analysis methods developed for instance in [S&F 83] and [Fla 88]. We show how these methods apply to the computation of the enumerative series related to the terms of an algebraic specification. We define the notion of *regular* rewriting systems, and consider cost series associated to operators that are described by such systems. We show how these analysis methods apply to compute such costs and provide an asymptotic evaluation of the average cost of an operator. Our results allow costs to be computed without any explicit manipulation of series. We provide the user with ready-to-use formulae, where the different parameters only depend on the "geometry" of the system, e.g. the number of constructors in the left handside of rules, number of occurrences of a derived operator in the right handside, etc.

Quantitative evaluation of rewriting systems had not yet been studied under such an approach (except in [CLR 80]), to our knowledge. From a different point of view, complexity of algebraic implementations has been studied in [BBWT 81, E&M 81, etc.] w.r.t. computability issues.

This paper is organised as follows : section 1 is devoted to illustrating on an example the use of complexity analysis methods on term rewriting systems ; section 2 presents the results on the enumerative series of term algebras that we use in the rest of the paper ; in section 3 we define regular systems and prove some of their properties ; in section 4 we show how the expression of the cost series for a "derived" operator can be systematically computed from the syntactic form of the rewrite rules ; we develop in section 5 the asymptotic evaluation of operator costs : we show conditions for the average cost to be asymptotically constant, polynomial or exponential, and give several examples.

1. Introductory example

This section is an introduction to complexity analysis methods by means of an example. We first show how to compute the enumerative series of binary trees constructed with two operators (a constant operator and a binary operator) ; we then give an asymptotic evaluation of the series coefficients. We then add an operator and describe its behavior by rewrite rules : we apply the analysis methods to the computation of this operator cost series and deduce its average cost.

Let us assume that one wants to evaluate the average cost of a given computation on some data set. The data belong to a set of objects, a size can be computed for each object ; let D_n denote the set of objects of size n, and N_n = card (D_n). Assuming all the objects have the same probability, the average cost is [see e.g. GSF 86] :

$$\overline{C}_n = \frac{1}{N_n} \sum_{d \in D_n} \text{cost}(d) = \frac{C_n}{N_n} .$$

Generating series are defined by associating the series $a(z) = \sum a_n z^n$ to a sequence (a_n) : to the sequence (N_n) is associated the *enumerative series* $N(z) = \sum N_n z^n$, and to the sequence (C_n) is associated the cost series $C(z) = \sum C_n z^n$. Computation of the coefficients of the generating series can be performed either using "exact" methods (e.g. using the Lagrange inversion theorem) or methods that provide an approximation, based on real or complex analysis techniques. The asymptotic value of the coefficients a_n of a complex series $\sum a_n z^n$ may be evaluated using results of complex functions theory (essentially based on Cauchy's formula) : the singularity closest to the origin determines the order of growth of the coefficients (more precisely, their exponential factor is determined by convergence radius of the series and their polynomial factor is function of the nature of the singularity) [see e.g. S&F 83].

Consider the example of a specification of binary trees with two constructors : the constant 'a' and the '_ . _' operator (think of "cons" in Lisp).
Considering the enumerative series : $N_{tree}(z) = \sum_{n \geq 0} N_n z^n$, since each n^{th} power of z appears as many times as there are trees of size n, we have : $N_{tree}(z) = \sum_{t \in T_{tree}} z^{|t|}$ where T_{tree} is the set of terms built with the constructors 'a' and '_ . _' .

Let us first perform the computation of $N_{tree}(z)$ by case analysis on terms :

$$N_{tree}(z) = \sum_{t = a} z^{|t|} + \sum_{t = t_1.t_2} z^{|t_1.t_2|} .$$

Since $|t_1 . t_2| = 1 + |t_1| + |t_2|$: $N_{tree}(z) = z + z \sum_{t_1 \in T_{tree}} z^{|t_1|} \sum_{t_2 \in T_{tree}} z^{|t_2|} = z\,(1 + N^2_{tree}(z))$.

In the general case, computation of $N_{tree}(z)$ can also be performed using systematic methods for computing enumerative series and cost series for algorithms on combinatorial structures [Fla 88] (in particular, one may apply these methods when trees are used as data structures to represent terms) ; the main steps of these methods are the following :

- take the construction primitives of the combinatorial object and deduce the structural equations ; in the case of tree this leads to :

$$\text{tree} = a + \underset{\text{tree}}{\overset{\cdot}{/\ \backslash}}\ \text{tree} \qquad \text{or} \qquad \text{tree} = a + . \times \text{tree} \times \text{tree}$$

where + is the disjoint union and × the cartesian product

• transpose the structural equation(s) to generating series, using the fact that, when considering the associated series, any disjoint union is expressed by a sum and any cartesian product is expressed by a product ; in our example, this leads to :

$$N_{tree}(z) = z + z\, N(z)\, N(z)$$

(the same result was obtained above by case analysis)

• solve the generating series equations. In the above example, simple resolution leads to : $N_{tree}(z) = \frac{1-\sqrt{1-4z^2}}{2z}$ (this solution is adequate since it is analytic at the origin), and further computation by series development leads $N_{2p} = 0$ and to $N_{2p+1} = \frac{1}{p+1}\binom{2p}{p}$ (Catalan numbers, cf. [Knu 73]).

Using the Stirling formula : $p! = \sqrt{2\pi p}\left[\frac{p}{e}\right]^p (1 + O(\frac{1}{p}))$, this yields :

$$N_{2p+1} = \frac{1}{\sqrt{\pi}} p^{-3/2} 2^{2p} (1 + O(\frac{1}{p}))$$

A more general approach is to use complex analysis methods (local analysis around singularities) for passing from functional equations over generating functions to asymptotic expressions of their coefficients. Continuing with our example, we have : $N_{tree}(z) = -\frac{\sqrt{1-2z}\,\sqrt{1+2z}}{2z} + \frac{1}{2z}$

$N_{tree}(z)$ is analytic for $|z| < \frac{1}{2}$ and has two singularities : $z = \frac{1}{2}$ and $z = -\frac{1}{2}$.

Let us apply the Newton expansion : $[z^n]\,(1-qz)^{\alpha}$ † $= q^n(-1)^n\binom{n}{\alpha} = q^n\binom{n-\alpha-1}{-\alpha-1}$

One shows that when $n \to +\infty$: $[z^n]\,(1-qz)^{\alpha} = \frac{n^{-\alpha-1}}{\Gamma(-\alpha)} q^n (1 + O(\frac{1}{n}))$

(where Γ is the Euler Gamma function : $\Gamma(x) = \int_0^{+\infty} e^{-t} t^{x-1} dt$. We recall that : $\Gamma(x+1) = x\,\Gamma(x)$, $\Gamma(1) = 1$ -- hence $\Gamma(n) = (n-1)!$ when $n \in N$ -- and $\Gamma(\frac{1}{2}) = \sqrt{\pi}$).

The contributions of singularities with same module are added together, leading in our case to :

$[z^n]\,N(z) = \frac{1}{\sqrt{2\pi}}\, 2^{n+1}\, n^{-3/2}\, (1 + O(\frac{1}{n}))$ for n odd, and 0 for n even.

A more systematic approach that uses "transfer lemmas" [Fla 88] is presented in section 2.

Now let us add a derived operator '_ ↑ _' (called "shuffle" since it is moving subtrees around) defined with the following rules :

$$(S_1) : a \uparrow t \to a\,.\,t, \quad (t_1\,.\,t_2) \uparrow a \to (t_1\,.\,t_2)\,.\,a, \quad (t_1\,.\,t_2) \uparrow (u_1\,.\,u_2) \to (t_1 \uparrow u_1)\,.\,(t_2 \uparrow u_2)$$

The cost of a term is the number of rewriting steps necessary to reduce it to its normal form for a given strategy ; the cost of an operator is defined as the cost of a term obtained by applying this operator to the terms in normal form. In all examples considered in this paper, it is insured that operator costs are independent from the evaluation strategy (cf. proposition in section 3).

Let us evaluate the cost functions for the operators of this specification : 'a' and '_ . _' being constructors, the corresponding cost functions are equal to zero. Computation of the cost function for the operation '_ ↑ _' will be done by means of the following generating function (cf. definition 4.1) :

$C^{\uparrow}(z) = \sum_{n \geq 0} C_n^{\uparrow} z^n$ where $C_n^{\uparrow} = \sum_{\substack{t,\, u \in T_{tree} \\ |t| + |u| = n}} \text{cost}\,(t \uparrow u)$ (where $|\,t\,|$ is the size of the term t, i.e. the total number of symbols that appear in t, and where T_{tree} is the set of terms in normal form, which is exactly here the set of terms built on the constructors 'a' and '_ . _').

† $[z^n]\,\Phi(z)$ denotes the coefficient of z^n in $\Phi(z)$

Hence : $$C^{\uparrow}(z) = \sum_{t,\, u \in T_{tree}} cost(t \uparrow u)\, z^{|t|+|u|}$$

Let us use a case analysis for terms in normal form to compute $C^{\uparrow}(z)$:

$$C^{\uparrow}(z) = \sum_{t \in T_{tree}} cost\,(a \uparrow t)\, z^{1+|t|} + \sum_{t_1,t_2 \in T_{tree}} cost\,((t_1.t_2) \uparrow a)\, z^{|t_1.t_2|+1}$$
$$+ \sum_{t_1,t_2,u_1,u_2 \in T_{tree}} cost\,((t_1 \,.\, t_2) \uparrow (u_1 \,.\, u_2))\, z^{|t_1.t_2|+|u_1.u_2|}$$

since : $cost\,(a \uparrow t) = 1 + cost\,(a \,.\, t) = 1,$

$cost\,((t_1.t_2) \uparrow a) = 1 + cost\,((t_1.t_2) \,.\, a) = 1,$

$cost\,((t_1 \,.\, t_2) \uparrow (u_1 \,.\, u_2)) = 1 + cost\,((t_1 \uparrow u_1).(t_2 \uparrow u_2))$

we have :

$$C^{\uparrow}(z) = z \sum_{t \in T_{tree}} z^{|t|} + z^2 \sum_{t_1 \in T_{tree}} z^{|t_1|} \sum_{t_2 \in T_{tree}} z^{|t_2|} + z^2 \sum_{t_1 \in T_{tree}} z^{|t_1|} \sum_{t_2 \in T_{tree}} z^{|t_2|} \sum_{u_1 \in T_{tree}} z^{|u_1|} \sum_{u_2 \in T_{tree}} z^{|u_2|}$$
$$+ z^2 \sum_{t_1,u_1,t_2,u_2 \in T_{tree}} cost\,((t_1 \uparrow u_1).(t_2 \uparrow u2))\, z^{|t_1|}\, z^{|u_1|}\, z^{|t_2|}\, z^{|u_2|}$$

since : $cost\,((t_1 \uparrow u_1).(t_2 \uparrow u_2)) = cost\,(t_1 \uparrow u_1) + cost\,(t_2 \uparrow u_2)$,

$$C^{\uparrow}(z) = z\, N_{tree}(z) + z^2\, N^2_{tree}(z) + z^2\, N^4_{tree}(z) + z^2 \sum_{t_1,u_1 \in T_{tree}} cost\,(t_1 \uparrow u_1)\, z^{|t_1|}\, z^{|u_1|} \sum_{t_2 \in T_{tree}} z^{|t_2|} \sum_{u_2 \in T_{tree}} z^{|u_2|}$$
$$+ z^2 \sum_{t_2,u_2 \in T_{tree}} cost\,(t_2 \uparrow u_2)\, z^{|t_2|}\, z^{|u_2|} \sum_{t_1 \in T_{tree}} z^{|t_1|} \sum_{u_1 \in T_{tree}} z^{|u_1|}$$
$$= N^2_{tree}(z)\ ^{\dagger} + 2\, z^2\, N^2_{tree}(z)\, C^{\uparrow}(z)$$

Hence : $$C^{\uparrow}(z) = \frac{N^2_{tree}(z)}{1 - 2\, z^2\, N^2_{tree}(z)}$$

Replacing in this expression $N_{tree}(z)$ by its value : $\frac{1-\sqrt{1-4z^2}}{2z}$ and using the same complex analysis method as for $N_{tree}(z)$ (development around singularities, Newton expansion, adding up the contributions of singularities) yields to : $C^{\uparrow}_{2p} = \frac{4}{\sqrt{\pi}} 2^{2p} p^{-3/2} (1 + O\,(\frac{1}{p}))$

Now the average cost is : $\overline{C}^{\uparrow}_{2p} = \frac{C^{\uparrow}_{2p}}{N_{2,2p}}$ where $N_{2,2p}$ (cf. section 2) is the number of tree couples (t_1, t_2) such that $|t_1| + |t_2| = 2p$. Computation of $N_{2,2p}$ yields :

$$\overline{C}^{\uparrow}_{2p} = 4(1 + O\,(\frac{1}{p}))$$

thus, the average number of rewriting steps when the $\uparrow$ operator is applied to a binary tree is asymptotically constant.

2. Enumerative series of the term algebra

We now apply to the term algebra some algebraic and analytic results on families of trees that have been developed in [M&M 78, S&F 83]. An enumerative series can be associated to the signature of a set of terms and complex analysis techniques can be used to extract asymptotic information on the series coefficients.

We denote by T_{Constr} the set of terms built on the signature Constr. Let us denote by α_k the number of symbols in Constr of arity k. We always suppose that $\alpha_0 \neq 0$.

---

† using the fact that : $N_{tree}(z) = z\,(1 + N^2_{tree}(z))$

Definition 2.1

Let N_n stand for the number of terms in T_{Constr} of size n. The *enumerative series* of T_{Constr} is :

$$N(z) = \sum_{n\geq 0} N_n z^n = \sum_{t\in T_{Constr}} z^{|t|}.$$

Let $\Phi(X)$ stand for the polynomial : $\Phi(X) = \sum_{k=0}^{p} \alpha_k X^k$, where 'p' denotes the largest arity in Constr, and where there exists $\alpha_k > 0$ with $k \geq 2$.†

We have the following results :

(1) N(z) is a solution of the functional equation : $N(z) = z \,.\, \Phi(N(z))$ (cf. section 1 : transposition from structural equations to generating series).

(2) Let us suppose that there is no polynomial Ψ and integer $d\geq 2$ such that $\Phi(X) = \Psi(X^d)$; let τ be the smallest root of the equation $\Phi(X) = X\Phi'(X)$, and $\rho = \frac{\tau}{\Phi(\tau)}$; ρ is the convergence radius of N(z) and $0 < \rho < 1$. It can be shown that τ is the only real positive root of the equation $\Phi(X) = X\Phi'(X)$ such that $|X| = \tau$. Moreover, $\tau = N(\rho)$ and ρ is the only singularity of N(z) such that $|z| = \rho$. Then around $z = \rho$ [M&M 78] :

$$N(z) = \tau - \sqrt{\frac{2\Phi(\tau)}{\Phi''(\tau)}}\left[1-\frac{z}{\rho}\right]^{1/2} + \gamma_1\left[1-\frac{z}{\rho}\right] + O\left(\left|1-\frac{z}{\rho}\right|^{3/2}\right) \qquad (F1)$$

(3) **Transfer Lemmas** : when it is possible to have an asymptotic development of a series around the singularity that is closest to the origin, under some conditions of analytic continuation (that are always fulfilled in the case of term algebras) an estimation of the series coefficients can be deduced from the series estimation through transfer lemmas [Fla 88].

We use the transfer lemmas in the following case :

Let $f(z) = h(z) + O(g(z))$ be the expansion of f around the singularity ρ, where h and g are standard functions of the type $\left[1 - \frac{z}{\rho}\right]^{\alpha}$ and h is of higher order than g around ρ. Then : $[z^n]\, f(z) = [z^n]\, h(z) + O([z^n]\, g(z))$.

Considering the expression (F1) and applying the transfer lemmas, one gets :

$$N_n = -\sqrt{\frac{2\Phi(\tau)}{\Phi''(\tau)}}\,\frac{\rho^{-n}\, n^{-3/2}}{\Gamma(-1/2)} + O\left[\rho^{-n} n^{-5/2}\right] \qquad \text{with } \Gamma(-1/2) = -2\sqrt{\pi}$$

i.e. finally :

$$N_n = \sqrt{\frac{\Phi(\tau)}{2\pi\Phi''(\tau)}}\,\rho^{-n}\, n^{-3/2}\,\left(1 + O\left(\frac{1}{n}\right)\right).$$

Note 1 :

For average cost computations, we need to evaluate the quantity $N_{m,n}$, that is the number of terms $t_1,\ldots,t_m \in T_{Constr}$ such that $|t_1|+\ldots+|t_m| = n$. Then :

$$\sum_{n=0}^{\infty} N_{m,n}\, z^n = \sum_{n=0}^{\infty} \sum_{\substack{t_1,\ldots,t_m\in T_{Constr} \\ |t_1|+\ldots+|t_m|=n}} z^n = \sum_{t_1,\ldots,t_m\in T_{Constr}} z^{|t_1|+\ldots+|t_m|} = (N(z))^m.$$

Thus, $N_{m,n}$ is the coefficient of z^n in $(N(z))^m$, and using (F1) we have :

$$(N(z))^m = \tau^m - m\,\tau^{m-1}\sqrt{\frac{2\Phi(\tau)}{\Phi''(\tau)}}\left[1-\frac{z}{\rho}\right]^{1/2} + \gamma_2\left[1-\frac{z}{\rho}\right] + O\left(\left|1-\frac{z}{\rho}\right|^{3/2}\right).$$

† If this condition is not satisfied, this means that $N(z) = z(1+\alpha_1 N)$, and thus : $N(z) = \frac{z}{1-\alpha_1 z}$

Now, using the transfer lemmas, we obtain :

$$N_{m,n} = m\,\tau^{m-1}\sqrt{\frac{\Phi(\tau)}{2\pi\Phi''(\tau)}}\,\rho^{-n}\,n^{-3/2}\;\left(1+O\left(\frac{1}{n}\right)\right)$$

and : $N_{m,n} = m\,\tau^{m-1}\,N_n = \frac{d}{dN}(N^m(z))\,|_{z=\rho}N_n$

More generally, given the polynomial on N(z) : $P(N(z)) = \sum_{i=1}^{q}\alpha_i N^i(z)$,

similarly : $[z^n]\,P(N(z)) = N_n\frac{dP}{dN}\,|_{z=\rho}$.

Note 2 :

The above results were obtained for the case where ρ was the only singularity of N(z) such that $|z| = \rho$. Let us now consider the case where there are more than one such singularity. If $\Phi(X) = \Psi(X^d)$, let τ still denote the unique real, positive root of the equation $\Phi(X) = X\Phi'(X)$. The other solutions of this equation are $e^{2ik\pi/d}\tau$, for k = 1,2,...,d-1. We now have :

$$N_n = d\sqrt{\frac{\Phi(\tau)}{2\pi\Phi''(\tau)}}\,\rho^{-n}\,n^{-3/2}\;\left(1+O\left(\frac{1}{n}\right)\right) \quad \text{if } n \equiv 1 \text{ [mod d]},$$
$$N_n = 0 \quad \text{otherwise.}$$

and :

$$N_{m,n} = d\,m\,\tau^{m-1}\sqrt{\frac{\Phi(\tau)}{2\pi\Phi''(\tau)}}\,\rho^{-n}\,n^{-3/2}\;\left(1+O\left(\frac{1}{n}\right)\right) \quad \text{if } n \equiv m \text{ [mod d]},$$
$$N_{m,n} = 0 \quad \text{otherwise.}$$

Example :

These results immediately apply to the example of section 1 where Constr = {a, _ . _ }. We have $\alpha_0 = 1$, $\alpha_1 = 0$, $\alpha_2 = 1$. Then $\Phi(X) = 1+X^2$, of the form $\Psi(X^2)$, with $\Psi(Y) = 1+Y$. τ satisfies : $1+\tau^2 = 2\tau^2$. Thus $\tau = 1$ and $\rho = \frac{\tau}{\Phi(\tau)} = 1/2.$, and we derive :

$$N_n = \frac{1}{\sqrt{2\pi}}\,2^{n+1}\,n^{-3/2}\;\left(1+O\left(\frac{1}{n}\right)\right) \text{ for n odd, and 0 for n even ;}$$

$$N_{m,n} = \frac{m}{\sqrt{2\pi}}\,2^{n+1}\,n^{-3/2}\;\left(1+O\left(\frac{1}{n}\right)\right) \text{ for } n \equiv m \text{ [mod d], and 0 otherwise.}$$

Notice that letting m = 2, n = 2p, we obtain : $N_{2,2p} = \frac{1}{\sqrt{2\pi}}\,2^{2p}\,p^{-3/2}\left(1+O\left(\frac{1}{p}\right)\right)$ which leads to the same result as obtained in section 1.

3. Regular systems

In this section we define regular systems and prove that they have the following properties : finite termination (or nœtherianity), confluence (i.e. unicity of normal form), and the number of rewriting steps to the normal form is independent of the strategy.

We suppose that the set of operator symbols Σ is partitioned into :

- a set Constr of *constructors* (these are functions that generate the set of terms T_{Constr}, and for which the generating function N was computed hereabove) ;
- a set Der of *derived operators* (that realize computations on terms of T_{Constr}).

We wish to ensure that any term of T_{Constr} (i.e. built with constructors only) is irreducible, and that for $f \in$ Der and for $t_1,...,t_n \in T_{Constr}$, $f(t_1,...,t_n)$ rewrites into a unique term of T_{Constr} in a finite amount of rewrite steps. To this effect, we are going to restrict the form of the rules that are acceptable.

Definition 3.1

A *Constr-enumeration* is a finite family of n-tuples $(\vec{\omega}_e)$ of $(T_{Constr}(X))^n$, where e belongs to a set of indices, such that :

- each $\vec{\omega}_e$ contains at least a constructor symbol
- for any $\vec{t} \in (T_{Constr})^n$, there exists a unique substitution $\sigma: X \to T_{Constr}$ and a unique e such that : $\vec{t} = \vec{\omega}_e\sigma$

Thus, intuitively speaking, the $(\vec{\omega}_e)$ form a description of $(T_{Constr}(X))^n$, that is at the same time exhaustive and non-ambiguous. For instance, with Constr={0, s}, we can take : $\vec{\omega}_1 = (x , 0)$ and $\vec{\omega}_2 = (x , s(y))$, as a Constr-enumeration of $(T_{Constr}(X))^2$ (here, $e \in \{1,2\}$) ; with the same set of constructors, $\vec{\omega}_1 = (0, 0)$, $\vec{\omega}_2 = (0, s(y))$, $\vec{\omega}_3 = (s(x), 0)$, $\vec{\omega}_4 = (s(x), s(y))$, is another example of a Constr-enumeration of $(T_{Constr}(X))^2$ (with $e \in \{1,2,3,4\}$). Let us note that the number of e's just reflects the degree of "detail" in the description.

Let X_e stand for the variables that appear in $\vec{\omega}_e$; in the previous example, we have :
$X_1 = \{\}$, $X_2 = \{y\}$, $X_3 = \{x\}$, $X_4 = \{x,y\}$.

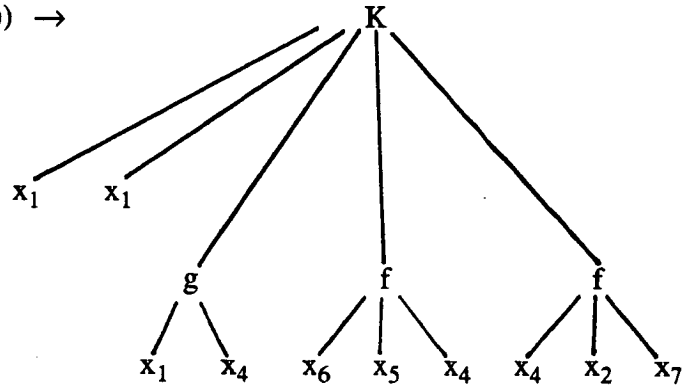
Definition 3.2

A *Constr-definition* of $f \in Der$ is a set of rewrite rules $R_f = (r_e)_{e \in D(f)}$, where D(f) is a set of indices for the rules R_f such that :

- each rule is of the form $r_e : f(\vec{\omega}_e) \to \rho_e$, where $(\vec{\omega}_e)_{e \in D(f)}$ is a Constr-enumeration of $T_{Constr}^{ar(f)}$ and ar(f) is the arity of f ;
- each ρ_e is of the form $\rho_e = K(x_1,...,x_n,\phi_1,...,\phi_m)$, where :
  - K is a context made of constructors only,
  - $\{x_1,...,x_n\} \subseteq X_e$, where $X_e = var(\vec{\omega}_e)$,
  - each ϕ_k is of the form $g(y_1,...,y_{ar(g)})$, where g is a derived operator, $\{y_1,...,y_{ar(g)}\} \subseteq X_e$, and $y_i \neq y_j$ if $i \neq j$.

A typical rule thus looks like :

$f(\vec{\omega}(x_1,x_2,x_3,x_4,x_5,x_6,x_7)) \to$



Definition 3.3

A set of R of rewrite rules is *regular* if it is of the form $R = \cup_{f \in Der} R_f$, where R_f is a Constr-definition of f, for each $f \in Der$

For instance, the following systems are regular :

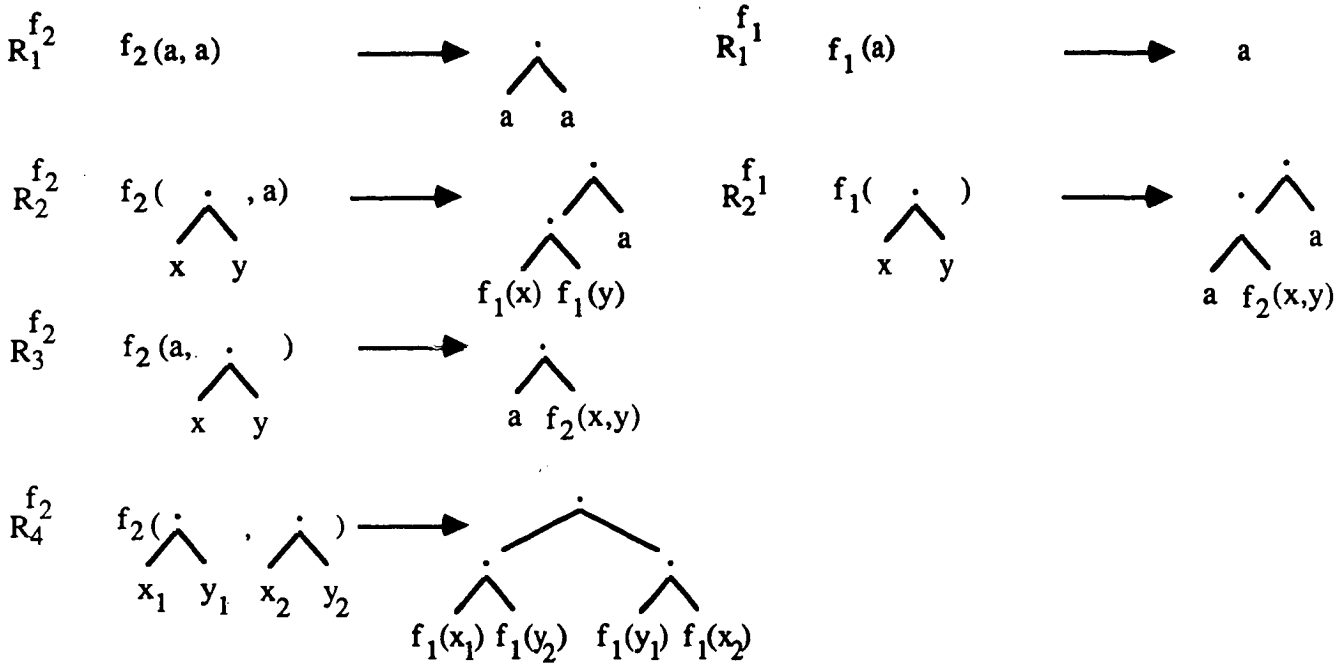
(S_2) Constr = {0, s}, Der = {+ , even} and

R_+ : $x + 0 \to x$, $x + s(y) \to s(x + y)$

R_{even} : $even(0) \to True$, $even(s(0)) \to False$, $even(s(s(x))) \to even(x)$

The Constr-enumeration associated to '+' is the one presented hereabove, while the one associated to 'even' is $\omega_1 = (0)$, $\omega_2 = (s(0))$, $\omega_3 = (s(s(x)))$.

(S_3) Constr = {a, _ . _ }, Der = {f_1(_), f_2(_)} and



Thus, regular systems can be mutually recursive.

We have the following result :

Theorem 3.1
A regular system is confluent and nœtherian. Moreover, it provides sufficiently complete and hierarchically consistent definitions of the derived operators w.r.t. the constructors.

Proof : confluence is because a regular system has no critical pair. Let us now prove nœtherianity. We order $T_{Constr \cup Der}$ by the *recursive path ordering* $>_{rpo}$ (cf. [Der 82,85]) such that all the constructors are equivalent, all the derived operators are equivalent, and the derived operators are greater than the constructors. Then consider a rule :

$$f(\vec{\omega}_i) \rightarrow K(x_1,...,x_n,\phi_1,...,\phi_m)$$

with the previous notations. Each ϕ_k is of the form $g(y_1,...,y_{ar(g)})$. Since by hypothesis $\vec{\omega}_i$ is not empty, the *multiset* $\{\vec{\omega}_i\}$ is strictly greater (for the associated ordering $\gg_{rpo}$) than the multiset $\{y_1,...,y_{ar(g)}\}$. Thus, $f(\vec{\omega}_i) >_{rpo} \phi_k$ for any k, and finally :

$$f(\vec{\omega}_i) >_{rpo} K(x_1,...,x_n,\phi_1,...,\phi_m)$$

This ends the proof of the termination of a regular system.
We notice that terms in T_{Constr} are irreducible, since no left-hand side admits a constructor at the root occurrence. This implies hierarchical consistence w.r.t. the constructors. Conversely, a term in normal form contains no derived operator, since otherwise a rule would apply to further reduce it. Thus, the system is also hierarchically complete.

In this article, we restrict attention to regular systems only. We then have the following result :

Proposition :
Let R be a regular system. For any term $t = f(t_1,...,t_n)$ with $f \in Der$ and $t_1,...,t_n \in T_{Constr}$, the number of rewrite steps between t and its normal from is independent of the rewriting strategy.

Proof : Let us denote by Γ the set of terms of $T_{Constr \cup Der}$ such that *at most one* derived operator appears from any path inside the term to the root. It is clear that, if R is regular, if $t \in \Gamma$ and $t \to^* t'$, then $t' \in \Gamma$. We are going to show, more generally, that for any $t \in \Gamma$, the cost of t does not depend of the evaluation strategy (which proves the previous lemma). *Ad absurdum*, suppose that this is not the case for a given t. We can write $t = K[f_1(\vec{\chi_1}),...,f_n(\vec{\chi_n})]$, where K and the $\vec{\chi_i}$'s are made of constructors only, and the f_i's are derived operators. We consider two cases :

Case 1 : if K is empty (and $t = f_1(\vec{\chi_1})$), then exactly one rule applies to t. We let $\phi(t)$ be the term such that $t \to \phi(t)$. Then, necessarily, the cost of $\phi(t)$ depends on the evaluation strategy.

Case 2 : else, if the cost of each $f_i(\vec{\chi_i})$ is equal to m_i, *whatever* the evaluation strategy, then the cost of t would be $m_1+...+m_n$, whatever the strategy, which would contradict the hypothesis. Thus, there exists an i such that the cost of $f_i(\vec{\chi_i})$ depends on the evaluation strategy. We then let $\phi(t) = f_i(\vec{\chi_i})$.

We define the ordering '>' by $t > t'$ *iff* either $t \to t'$ or t' is a strict subterm of t. '>' is well-founded. Now, the infinite chain :

$$t,\ \phi(t),\ \phi(\phi(t)),\ ...$$

is decreasing for '>', which yields the desired contradiction. This terminates the proof.

Note : the property is not true for elements outside of Γ. Consider for instance the regular system :

$$x \mid 0 \to 0, \qquad x \mid s(y) \to s(x \mid y)$$

The term '(0 | 0) | 0' (which is not in Γ) would be normalized in respectively 1 or 2 steps by an "outermost" or an "innermost" strategy.

4. Cost series for the derived operators in regular systems

This section and the following present the two main theorems of the paper. Theorem 4.1 gives the expression of the cost series in function of the syntactic form of the rewriting system, Theorem 5.1 gives an asymptotic equivalent of the average cost of a derived operator on a term of size n.

Definition 4.1

Let $f \in Der$ of arity m. Let C_n^f stand for sum the number of rewrite steps of the term $f(t_1,...,t_m)$ to its normal form, where the $(t_i)_{1 \le i \le m}$ range over T_{Constr} and are such that $|t_1| + ... + |t_m| = n$:

$$C_n^f = \sum_{\substack{t_1,...,t_m \in T_{Constr} \\ |t_1|+...+|t_m|=n}} cost(f(t_1,...,t_m))$$

The *cost series associated to f* is :

$$C^f(z) = \sum_{n \ge 0} C_n^f z^n = \sum_{t_1,...,t_m \in T_{Constr}} cost(f(t_1,...,t_m))\ z^{|t_1|+...+|t_m|}$$

From now on, we suppose that $Der = \{f_1,...,f_{NDer}\}$ where NDer is the number of derived operators. Given a regular system, with each f_i defined by a Constr-definition (see definition 3.2), we can write :

$$C^{f_i}(z) = \sum_{t_1,...,t_{ar(f_i)} \in T_{Constr}} cost(f_i(t_1,...,t_{ar(f_i)}))\ z^{|t_1|+\cdots+|t_{ar(f_i)}|} = \sum_{e \in D(f_i)} \sum_{\sigma} cost(f_i(\vec{\omega}_e^i \sigma))\ z^{|\vec{\omega}_e^i \sigma|}.$$

where $(\vec{\omega}_e^i)_{e \in D(f_i)}$ is a Constr-enumeration of $T_{Constr}^{ar(f_i)}$.

We have : $cost(f_i(\vec{\omega}_e^i \sigma)) = 1 + \sum_{1 \le k \le n_e^i} cost(\phi_{k,i,e} \sigma)$, in accordance with definition 3.2. Thus :

$$C^{f_i}(z) = \underbrace{\sum_{e \in D(f_i)} \sum_{\sigma} z^{|\vec{\omega}_e^i \sigma|}}_{<\text{-------A-------}>} + \underbrace{\sum_{e \in D(f_i)} \sum_{\sigma} \sum_{1 \le k \le n_e} cost(\phi_{k,i,e} \sigma)\ z^{|\vec{\omega}_e^i \sigma|}}_{<\text{----------B----------}>}.$$

Where $A = N^{ar(f_i)}(z)$ is a constant part of this sum, and B is a recursive part.

In order to simplify the B part of C^{f_i}, we first notice that it is actually quantified over e corresponding to the rules with non-constant right handsides, that we denote $D_{nc}(f_i)$. Let $X_{e,i}$ stand for the variables of $\vec{\omega}_e^i$. Then, we may write $\phi_{k,i,e} = f_j(y_1,...,y_{ar(f_j)})$, for a certain j.

We suppose that σ restricted to the variables $X_{e,i}$ of $\vec{\omega}_e^i$ is defined by :

$\{\sigma(y_1)= t_1,...,\sigma(y_{ar(f_j)})= t_{ar(f_j)}$, and $\sigma(w_i)= t'_i$, for all $w_i \in X_{e,i} - \{y_i\}$ $\}$

the w_i's are the variables that appear in the left handsides and not in the right handsides of the rule f_e^i, and we let l(i,j,e) be the number of w_i's.

Let $\xi_{e,i}$ stand for the number of constructors appearing in $\vec{\omega}_e^i$ (let us recall that, according to Definition 3.1, $\xi_e \neq 0$).

We have : $|\vec{\omega}_e^i\sigma| = \xi_{e,i} + |t_1|+...+|t_{ar(f_j)}| + |t'_1|+...+|t'_{l(i,j,e)}|$

Then :

$$\sum_{e\in D(f_i)} cost(\phi_{k,i,e}\sigma)\, z^{|\vec{\omega}_e^i\sigma|} = z^{\xi_{e,i}} \sum_{e\in D_{nc}(f_i)} cost(f_j(t_1,...,t_{ar(f_j)}))\, z^{|t_1|+\cdots+|t_{ar(f_j)}|}\, z^{|t'_1|+\cdots+|t'_{l(i,j,e)}|}.$$

Let $\varepsilon_{e,j}^i$ stand for the number of occurrences of 'f_j' in the right handside of the rule r_e^i. The B part of C^{f_i} finally rewrites into :

$$\sum_{e\in D_{nc}(f_i)} z^{\xi_{e,i}} \sum_{1\le j\le NDer} \varepsilon_{e,j}^i \sum_{t_1,...,t_{ar(f_j)}\in T_{Constr}} cost(f_j(t_1,...,t_{ar(f_j)}))\, z^{|t_1|+\cdots+|t_{ar(f_j)}|}\, z^{|t'_1|+\cdots+|t'_{l(i,j,e)}|} =$$

$$\sum_{e\in D_{nc}(f_i)} \sum_{1\le j\le NDer} \varepsilon_{e,j}^i\, z^{\xi_{e,i}}\, N^{l(i,j,e)}\, C^{f_j}(z) = \sum_{e\in D_{nc}(f_i)} \sum_{1\le j\le m} \varepsilon_{e,j}^i\, z^{\xi_{e,i}}\, N^{|X_{e,i}|-ar(f_j)}\, C^{f_j}(z).$$

Going back to the example in section 1, we have Constr = {a, .}, the rule system is :

(S_1) : $a \uparrow t \to a \,.\, t$, $(t_1 \,.\, t_2) \uparrow a \to (t_1 \,.\, t_2) \,.\, a$, $(t_1 \,.\, t_2) \uparrow (u_1 \,.\, u_2) \to (t_1 \uparrow u_1) \,.\, (t_2 \uparrow u_2)$

and we have : $\vec{\omega}_1^{\uparrow} = (a, t)$, $\vec{\omega}_2^{\uparrow} = (t_1 \,.\, t_2, a)$, $\vec{\omega}_3^{\uparrow} = (t_1 \,.\, t_2, u_1 \,.\, u_2)$, $X_{1,\uparrow} = \{t\}$, $X_{2,\uparrow} = \{t_1, t_2\}$, $X_{3,\uparrow} = \{t_1, t_2, u_1, u_2\}$, $\xi_{1,\uparrow} = 1$, $\xi_{2,\uparrow} = 2$, $\xi_{3,\uparrow} = 2$.

$\varepsilon_{3,\uparrow}^{\uparrow} = 2$ *(the others are null),* $\xi_{3,\uparrow} = 2$, $|X_{3,\uparrow}| = 4$, *and* $ar(\uparrow) = 2$, *which leads to the result :* $C^{\uparrow}(z) = N_{tree}^2(z) + 2\, z^2\, N_{tree}^2(z)\, C^{\uparrow}(z)$.

Back to our general development, let $\vec{C}(z)$ stand for the vector $\begin{bmatrix} C^{f_1}(z) \\ ... \\ C^{f_{NDer}}(z) \end{bmatrix}$ and $\vec{Y}(z)$ for the vector $\begin{bmatrix} N^{ar(f_1)}(z) \\ ... \\ N^{ar(f_{NDer})}(z) \end{bmatrix}$. We also define $M_{i,j}(z) = \sum_{e\in D_{nc}(f_i)} \varepsilon_{e,j}^i\, z^{\xi_{e,i}}\, N^{|X_{e,i}|-ar(f_j)}(z)$,

with : $\varepsilon_{e,j}^i$ the number of occurences of the 'f_j' in the right handside of the rule r_e^i (i.e. the e-th rule defining f_i), $\xi_{e,i}$ the number of constructors appearing in $\vec{\omega}_e^i$, $|X_{e,i}|$ the number of variables in $\vec{\omega}_e^i$. We let M(z) denote the matrix $(M_{i,j}(z))_{1\le i,j\le NDer}$. We obtain the central result of this paper :

Theorem 4.1

The cost series satisfies the equation : $\vec{C}(z) = M(z)\,\vec{C}(z) + \vec{Y}(z)$. The expression of each cost series is :

$$C^{f_i}(z) = \frac{\det(\,(Id-M)^{[i]}(z)\,)}{\det(\,Id-M(z)\,)},$$

where Id is the identity matrix, and $(Id-M)^{[i]}(z)$ is the matrix Id-M(z), the i^{th} column of which being replaced by $\vec{Y}(z)$.

Since $z = \frac{N(z)}{\Phi(N(z))}$, each $C^{f_i}(z)$ may be rewritten into the following form : $C^{f_i}(z) = \frac{P^i(N(z))}{Q^i(N(z))}$, where P and Q are respectively prime polynomials with integer coefficients. Now, in order to evaluate

the $C_n^{f_i}$, we have to determine the smallest singularity of each $C^{f_i}(z)$. Its singularities are :

- either the singularities of N(z), the smallest being for $z = \rho$,
- or the z's such that $Q^i(N(z)) = 0$.

5. Asymptotic evaluation of the operator costs

Let us denote by ρ_0^i the smallest real positive root of $Q^i(N(\rho_0^i)) = 0$ (with the convention that $\rho_0^i = \infty$ if $Q^i(N(z))$ has no root for $|z| \leq \rho_0^i$). We now have the following main theorem :

Theorem 5.1

The average cost satisfies the following asymptotic equations :

(1) if $\rho < \rho_0^i$, then : $\quad \overline{C}_n^{f_i} = k_1 \ (1 + O(\frac{1}{n}))$

(2) if $\rho = \rho_0^i$, then: $\quad \overline{C}_n^{f_i} = k_2 n^{q/2} \ (1+O(\frac{1}{\sqrt{n}}))$

(3) if $\rho > \rho_0^i$, then : $\quad \overline{C}_n^{f_i} = k_3 \left[\frac{\rho}{\rho_0^i}\right]^n n^{r+1/2} \ (1+O(\frac{1}{n}))$

The k_i's are real numbers, and q and r are strictly positive integers ; all of them can be expressed simply (as shown hereafter - Results 5.1, 5.2, and 5.3).

Let us note that given the parameters depending on the geometry of the rewriting system, automatic computation of M(z), Y(z) and $C^{f_i}(z)$ can be performed with the assistance of a formal computation system (in the case of our examples, we used the MAPLE system [MAPLE 85]). In order to compute the mean cost, the only difficult point may be to compute the zeroes of the denominator (computation of the singularity of N(z) should not be problematic since constructors arity is usually less than or equal to 3).

We now proceed to proving theorem 5.1 by considering successively the three cases.

Study of case (1)

We can write, using Taylor expansion formula around τ :

$$\frac{P_i(N(z))}{Q_i(N(z))} = \frac{P_i(\tau)}{Q_i(\tau)} + \frac{d}{dN}\left[\frac{P_i}{Q_i}\right]_{|N=\tau} (N(z)-\tau) + \frac{d^2}{dN^2}\left[\frac{P_i}{Q_i}\right]_{|N=\tau} \frac{(N(z)-\tau)^2}{2!} + O(|N(z)-\tau|^2).$$

Applying the transfer lemmas of section 2, and using the approximation :

$$N(z) = \tau - \sqrt{\frac{2\Phi(\tau)}{\Phi''(\tau)}}\left[1-\frac{z}{\rho}\right]^{1/2} + \gamma 1 \left[1-\frac{z}{\rho}\right] + O\left(\left|1-\frac{z}{\rho}\right|^{3/2}\right) \qquad (F1)$$

we get :

$$C_n^{f_i} = k_1 \sqrt{\frac{\Phi(\tau)}{2\pi\Phi''(\tau)}} \rho^{-n} n^{-3/2} \ (1 + O(\frac{1}{n})), \quad \text{with } k_1 = \frac{d}{dN}\left[\frac{P_i}{Q_i}\right]_{|N=\tau}$$

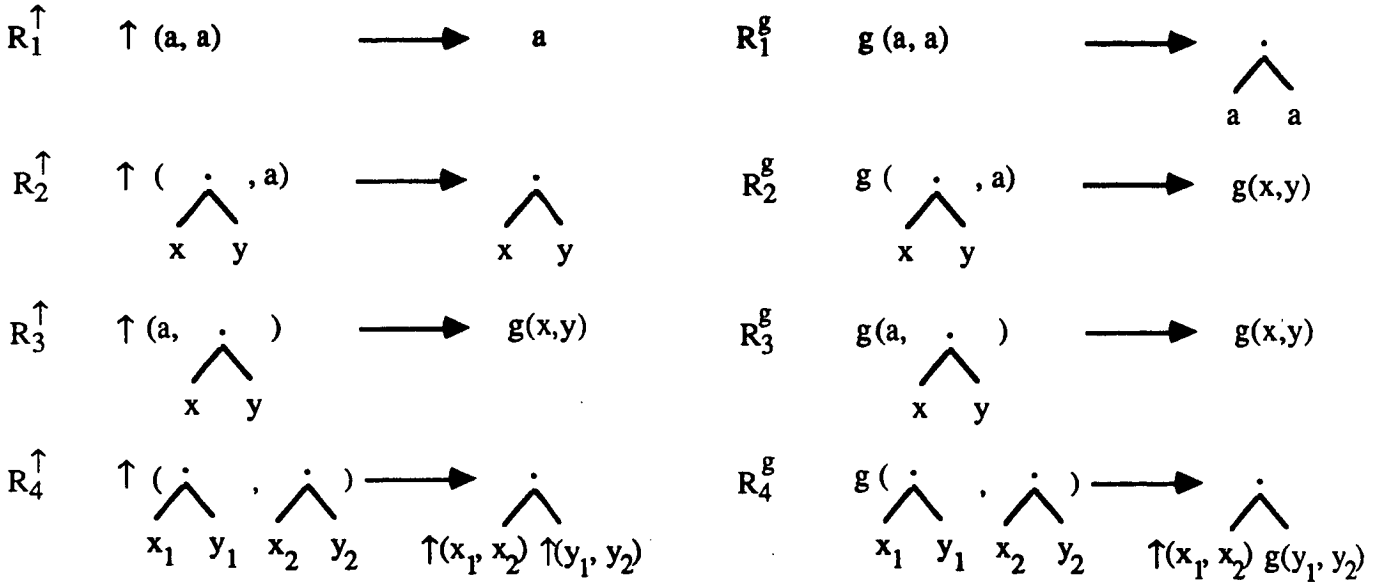
(when there are d singularities on the circle of convergence of N(z), k_1 is the above expression multiplied by d if $n \equiv ar(f_i)$ [mod d], and $C_n^{f_i} = 0$ otherwise), and finally :

Result 5.1

$$\overline{C}_n^{f_i} = \frac{C_n^{f_i}}{N_{m,n}} = k_1 \ (1 + O(\frac{1}{n})), \quad \text{with } k_1 = \frac{1}{ar(f_i)\tau^{ar(f_i)-1}} \frac{d}{dN}\left[\frac{P_i}{Q_i}\right]_{|N=\tau} \qquad \square$$

Example 5.1

We consider, on binary trees built as previously (cf. section 1 and example in section2), another version of a shuffle function on trees, defined by the following set of rules :



Let us recall that $N(z) = z\,(1 + N^2(z))$, $\rho = 1/2$ and $\tau = N(\rho) = 1$. The Constr-enumerations used are the same for both operations :

$\omega_1^{\uparrow} = \omega_1^{g} = (a, a)$ $\quad \xi_{1,\uparrow} = \xi_{1,g} = 2$ $\quad |X_{1,\uparrow}| = |X_{1,g}| = 2$

$\omega_2^{\uparrow} = \omega_2^{g} = (x.y, a)$ $\quad \xi_{2,\uparrow} = \xi_{2,g} = 2$ $\quad |X_{2,\uparrow}| = |X_{2,g}| = 2$

$\omega_3^{\uparrow} = \omega_3^{g} = (a, x.y)$ $\quad \xi_{3,\uparrow} = \xi_{3,g} = 2$ $\quad |X_{3,\uparrow}| = |X_{3,g}| = 2$

$\omega_4^{\uparrow} = \omega_4^{g} = (x_1 \cdot y_1, x_2 \cdot y_2)$ $\quad \xi_{4,\uparrow} = \xi_{4,g} = 2$ $\quad |X_{4,\uparrow}| = |X_{4,g}| = 4$

The matrix M(z) associated to the ↑ and g is : $M(z) = \begin{bmatrix} 2z^2N^2(z) & z^2 \\ z^2N^2(z) & 2z^2+z^2N^2(z) \end{bmatrix}$, the first line of M(z) relates to the definition of ↑ : $\varepsilon_{4,\uparrow}^{\uparrow} = 2$, and $\varepsilon_{3,g}^{\uparrow} = 1$; the second line of M(z) relates to the definition of g : $\varepsilon_{4,\uparrow}^{g} = 1$ for the first element, and $\varepsilon_{2,g}^{g} = \varepsilon_{3,g}^{g} = 1$ and $\varepsilon_{4,g}^{g} = 1$ for the second element.

We also have $Y(z) = \begin{bmatrix} N^2(z) \\ N^2(z) \end{bmatrix}$. This yields, after computation of the determinants, and replacement of z by $\dfrac{N(z)}{1+N^2(z)}$:

$$C^{\uparrow}(z) = \frac{N^2(z)\,(1+N^2(z))^3}{1+2N^2(z)-N^4(z)-N^6(z)} = \frac{P_1(N(z))}{Q_1(N(z))}$$

$$C^{g}(z) = \frac{(1+2N^2(z))\,N^2(z)\,(1+N^2(z))^2}{1+2N^2(z)-N^4(z)-N^6(z)} = \frac{P_2(N(z))}{Q_2(N(z))}$$

The denominator $(1+2N^2-N^4-N^6)$ has no root for $N \in [0,1]$. We are therefore in the current case (1).

Computation gives : $\dfrac{d}{dN}\left[\dfrac{P_1}{Q_1}\right]_{|N=1} = 88$ and $\dfrac{d}{dN}\left[\dfrac{P_2}{Q_2}\right]_{|N=1} = 136.$

Finally : $\overline{C}_n^{\uparrow} = 44\,(1 + O(\frac{1}{n}))$ $\qquad$ $\overline{C}_n^{g} = 68\,(1 + O(\frac{1}{n}))$.

End of example 5.1

Study of case (2)

We can write :

$$\frac{P_i(N(z))}{Q_i(N(z))} = \frac{1}{(N(z)-\tau)^s}\frac{P_i(N(z))}{\overline{Q}_i(N(z))},$$

where s is a strictly positive integer, and $\overline{Q}_i$ is an integer polynomial such that $\overline{Q}_i(\tau) \neq 0$.

Then, around τ : $\frac{P_i(N(z))}{Q_i(N(z))} = \frac{P_i(\tau)}{\overline{Q}_i(\tau)}(N(z)-\tau)^{-s} + \gamma_2(N(z)-\tau)^{-s+1} + O(|N(z)-\tau|^{-s+2})$.

Applying the transfer lemmas of section 2, and using the approximation (F1), we obtain :

$$C_n^{f_i} = k'_2\,\rho^{-n}n^{\frac{s}{2}-1}\left(1+O(\frac{1}{n^{1/2}})\right), \text{ with } k'_2 = (-1)^s\frac{P_i(\tau)}{\overline{Q}_i(\tau)}\left[\frac{2\Phi(\tau)}{\Phi''(\tau)}\right]^{-\frac{s}{2}}\frac{1}{\Gamma(s/2)}$$

(when there are d singularities on the circle of convergence of N(z), k'_2 is the above expression multiplied by d if $n \equiv ar(f_i)$ [mod d], and $C_n^{f_i} = 0$ otherwise), and thus :

$$\overline{C}_n^{f_i} = k_2 n^{\frac{s+1}{2}}\left(1+O(\frac{1}{n^{1/2}})\right), \text{ with } k_2 = (-1)^{s-1}\frac{1}{ar(f_i)\tau^{ar(f_i)-1}}\frac{P_i(\tau)}{\overline{Q}_i(\tau)}\left[\frac{2\Phi(\tau)}{\Phi''(\tau)}\right]^{-\frac{(s+1)}{2}}\frac{\Gamma(-1/2)}{\Gamma(s/2)}.$$

Finally, using $\Gamma(1/2) = \sqrt{\pi}$ and $\Gamma(s/2) = \begin{cases} (p-1)! & \text{if } s=2p \\ \frac{(2p)!}{4^p p!}\sqrt{\pi} & \text{if } s=2p+1 \end{cases}$, we obtain :

Result 5.2

$$\overline{C}_n^{f_i} = k_2 n^{\frac{s+1}{2}}\left(1+O(\frac{1}{n^{1/2}})\right),$$

with

$$k_2 = -\frac{1}{ar(f_i)\tau^{ar(f_i)-1}}\frac{P_i(\tau)}{\overline{Q}_i(\tau)}\left[\frac{2\Phi(\tau)}{\Phi''(\tau)}\right]^{-(p+1)}2^{2p+1}\frac{p!}{(2p)!} \quad \text{if } s = 2p+1$$

$$k_2 = \frac{1}{ar(f_i)\tau^{ar(f_i)-1}}\frac{P_i(\tau)}{\overline{Q}_i(\tau)}\left[\frac{2\Phi(\tau)}{\Phi''(\tau)}\right]^{-\frac{2p+1}{2}}\frac{2\sqrt{\pi}}{(p-1)!} \quad \text{if } s = 2p \qquad \square$$

Example 5.2a

We consider the same system as in example 5.1, except that rule $R_{\uparrow 2}$ is replaced by rule $R_{\uparrow 2'}$:

$R_{2'}^{\uparrow}$ $\uparrow$ ($\overset{x}{\cdot}$ $\overset{y}{\cdot}$, a) ⟶ g(x,y)

The matrix M(z) associated to the $\uparrow$ and g is now : $\begin{bmatrix} 2z^2N^2(z) & 2z^2 \\ z^2N^2(z) & 2z^2+z^2N^2(z) \end{bmatrix}$.

We obtain, after replacement of z by $\frac{N(z)}{1+N^2(z)}$: $\det(I-M(z)) = (1-N(z))\frac{(2N^2(z)+1)(N(z)+1)}{(1+N^2(z))^3}$

which yields :

$$C^{\uparrow}(z) = \frac{1}{1-N(z)}\frac{N^2(z)(1+N^2(z))}{1+N(z)} = \frac{1}{1-N(z)}\frac{P_1(N(z))}{\overline{Q}_1(N(z))}$$

$$C^{g}(z) = \frac{1}{1-N(z)}\frac{N^2(z)(1+N^2(z))}{1+N(z)} = \frac{1}{1-N(z)}\frac{P_2(N(z))}{\overline{Q}_2(N(z))}$$

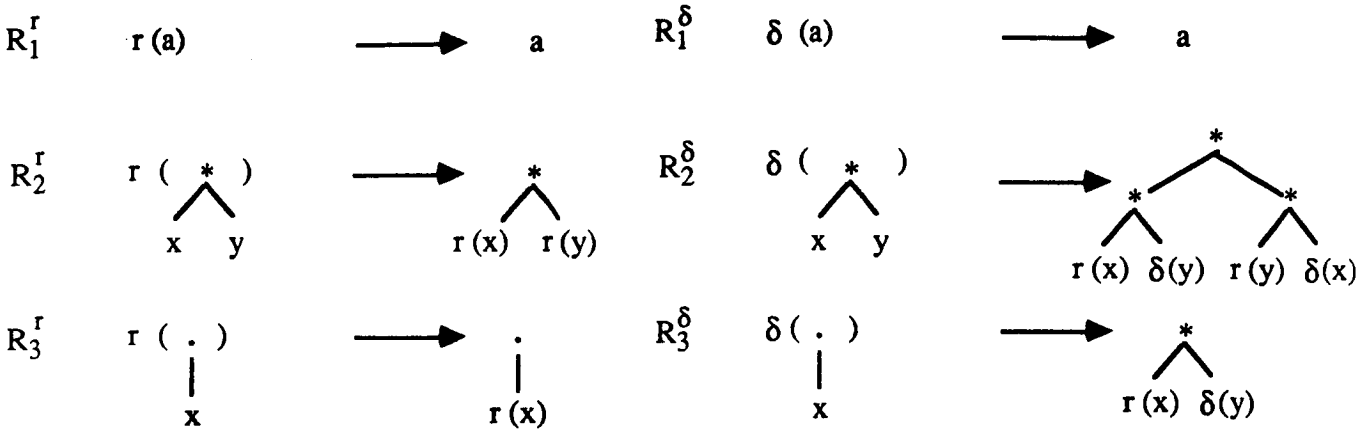
We are in the current case (2). Computation gives : $\frac{P_1(1)}{\overline{Q}_1(1)} = \frac{P_2(1)}{\overline{Q}_2(1)} = 1$, and finally :

$$\overline{C}_n^{\uparrow} = \overline{C}_n^{g} = \frac{n}{2}\,(1+O(\frac{1}{\sqrt{n}}))$$

End of example 5.2a

Example 5.2b

Let us consider a system where the constructors are Constr = {a, *, .} and the derived operators Der = {δ, r} defined by the following rules :



Since ar(.) = 1 and ar(*) = 2, $N(z) = z\,(1 + N(z) + N^2(z))$, which yields $\rho = 1/3$ and $\tau = 1$.

We have $M(z) = \begin{bmatrix} 2\,z\,N(z) + z & 2\,z\,N(z) + z \\ 0 & 2\,z\,N(z) + z \end{bmatrix}$ and $Y(z) = \begin{bmatrix} N(z) \\ N(z) \end{bmatrix}$.

We then get : $C^{\delta}(z) = \dfrac{N(z)}{(1 - z - 2z\,N(z))^2} = \dfrac{N(z)\,(1 + N(z) + N^2(z))^2}{(1 - N(z))^2\,(1 + N(z))^2}$

and $C^{r}(z) = \dfrac{N(z)}{1 - z - 2z\,N(z)} = \dfrac{N(z)\,(1 + N(z) + N^2(z))}{(1 - N(z))\,(1 + N(z))}$.

Finally : $\overline{C}_n^{\delta} = \dfrac{\sqrt{3\pi}}{2} n^{3/2}(1+O(\dfrac{1}{\sqrt{n}}))$ and $\overline{C}_n^{r} = n\,(1+O(\dfrac{1}{\sqrt{n}}))$.

Study of case (3)

Let $\tau_0^i = N(\rho_0^i)$. We have : $0 < \rho_0^i < \rho$, and $0 < \tau_0^i < \tau$. We can write :

$$\frac{P_i(N(z))}{Q_i(N(z))} = \frac{1}{(N(z)-\tau_0^i)^s}\frac{P_i(N(z))}{\overline{Q}_i(N(z))},$$

where s is a strictly positive integer, and $\overline{Q}_i$ is an integer polynomial such that $\overline{Q}_i(\tau_0^i) \neq 0$. Then, using a Taylor expansion of $z = \dfrac{N(z)}{\Phi(N(z))}$ in the neighbourhood of ρ_0^i, we get :

$$z = \frac{N(z)}{\Phi(N(z))} = \frac{\tau_0^i}{\Phi(\tau_0^i)} + \frac{d}{dN}\left[\frac{N}{\Phi(N)}\right]_{|N=\tau_0^i}(N-\tau_0^i) + O(|N-\tau_0^i|^2),$$

from which we derive :

$$N - \tau_0^i = \frac{1}{\dfrac{d}{dN}\left[\dfrac{N}{\Phi(N)}\right]_{|N=\tau_0^i}}(z-\rho_0^i) + O(|z-\rho_0^i|^2) = \frac{-1}{\dfrac{1}{\tau_0^i} - \dfrac{\Phi'(\tau_0^i)}{\Phi(\tau_0^i)}}(1-\frac{z}{\rho_0^i}) + O(|1-\frac{z}{\rho_0^i}|^2).$$

Using this development in the previous expression of $\dfrac{P_i(N(z))}{Q_i(N(z))}$ yields :

$$\frac{P_i(N(z))}{Q_i(N(z))} = (-1)^s(1-\frac{z}{\rho_0^i})^{-s}\left[\frac{1}{\tau_0^i} - \frac{\Phi'(\tau_0^i)}{\Phi(\tau_0^i)}\right]^s \frac{P_i(\tau_0^i)}{\overline{Q}_i(\tau_0^i)} + O\left(\left|1-\frac{z}{\rho_0^i}\right|^{-s+1}\right).$$

Thus,

$C_n^{f_i} = (-1)^s \left[\frac{1}{\tau_0^i} - \frac{\Phi'(\tau_0^i)}{\Phi(\tau_0^i)}\right]^s \frac{P_i(\tau_0^i)}{\overline{Q}_i(\tau_0^i)} (\rho_0^i)^{-n} \frac{n^{s-1}}{\Gamma(s)} \quad (1+O(\frac{1}{n}))$, (when there are d singularities on the circle of convergence of N(z), $C_n^{f_i}$ is the above expression multiplied by d if $n \equiv ar(f_i)$ [mod d], and $C_n^{f_i} = 0$ otherwise),

and finally :

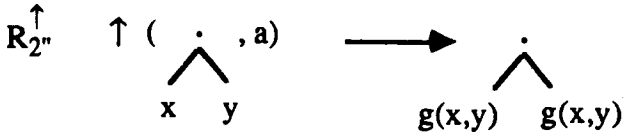
Result 5.3

$$\overline{C}_n^{f_i} = k_3 \left[\frac{\rho}{\rho_0^i}\right]^n n^{s+\frac{1}{2}} (1+O(\frac{1}{n})),$$

$$\text{with } k_3 = \frac{(-1)^s}{(s-1)!} \frac{1}{ar(f_i)\tau^{ar(f_i)-1}} \left[\frac{1}{\tau_0^i} - \frac{\Phi'(\tau_0^i)}{\Phi(\tau_0^i)}\right]^s \frac{P_i(\tau_0^i)}{\overline{Q}_i(\tau_0^i)} \sqrt{\frac{2\pi\Phi''(\tau)}{\Phi(\tau)}} \quad \square$$

Example 5.3

We consider the same system as in example 5.1 (or example 5.2a), except that rule $R_{\uparrow 2}$ (or rule $R_{\uparrow 2'}$) is replaced by rule $R_{\uparrow 2''}$:



The matrix M(z) associated to the ↑ and g is now :

$$\begin{bmatrix} 2z^2N^2(z) & 3z^2 \\ z^2N^2(z) & 2z^2+z^2N^2(z) \end{bmatrix}.$$

This yields, after computation of the determinants, and replacement of z by $\frac{N(z)}{1+N^2(z)}$:

$$C^{\uparrow}(z) = \frac{(1+N^2(z))^2\,(1+3N^2(z))\,N^2(z)}{1+2N^2(z)-N^4(z)-3N^6(z)}$$

$$C^{g}(z) = \frac{(1+N^2(z))^2\,(1+2N^2(z))\,N^2(z)}{1+2N^2(z)-N^4(z)-3N^6(z)}$$

The expression $(1+2N^2(z)-N^4(z)-3N^6(z))$ admits a root for $\tau_0 \sim 0.93336$, which gives $\rho_0 \sim 0.49881$ ($< \rho = 1/2$). We are therefore in the current case (3), and computation gives finally :

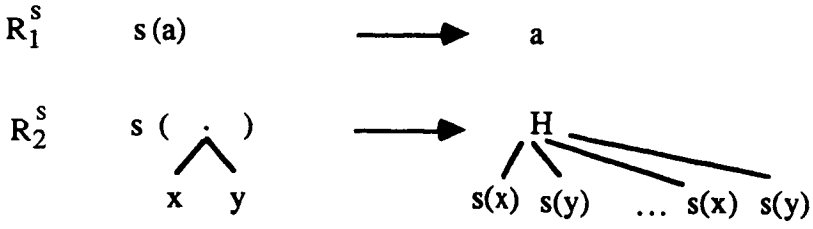
$$\overline{C}_n^{\uparrow} = k_3 \left[\frac{\rho}{\rho_0^i}\right]^n \sqrt{n}\ (1+O(\frac{1}{\sqrt{n}})) \qquad \overline{C}_n^{g} = k'_3 \left[\frac{\rho}{\rho_0^i}\right]^n \sqrt{n}\ (1+O(\frac{1}{\sqrt{n}}))$$

with $k_3 \sim 0.27901$, $k'_3 \sim 0.21234$, and $\frac{\rho}{\rho_0^i} \sim 1.00238$

End of example 5.3

We notice that, simply modifying *one* rule between examples 5.1, 5.2a and 5.3, induces respectively constant, polynomial or exponential cost. This illustrates the great sensitivity of the cost of rewriting w.r.t. mild modifications within the rewrite rules.

In the case of example 5.3, ρ/ρ_0 is close to 1, but the exponential growth can be much faster ; let us consider the following example on trees with the same constructors as in example 5.3 :



where H is the complete binary tree with 16 leaves labelled by s(x) or s(y).

We have : $C^s(z) = \dfrac{N(z)}{1 - 16\, z\, N(z)} = \dfrac{N(z)\,(1 + N^2(z))}{1 - 15\, N^2(z)}$.

The denominator is null for $\tau_0 = 1/\sqrt{15}$ that is $\rho_0 = \sqrt{15}/16 \sim 3.888/16$, so that $\rho/\rho_0 \sim 2$.

Note : computations for the examples in this section were also checked using the assistant algorithm analyser LUO [FSZ 88].

6. Conclusion

For the class of the *regular* term rewriting systems, we have provided ways of obtaining asymptotic evaluations of the cost series. The user does not need to actually manipulate formal series, since our results are given under the form of ready-to-use formulae. These results solely depend on physical characteristics of the system, easily obtainable : number of variables and of constructors in the left-hand sides, occurrences of derived operators in the right-hand sides. Then, the average cost is constant, polynomial or exponential, according to the position of the singularity of the expressions $Q_i(N(z))$ closest to the origin. Such an analysis can be performed automatically on the basis of a formal computation system.

Acknowledgements

We thank Jean-Pierre Jouannaud for contribution to the proof of Theorem 3.1 and Philippe Flajolet for fruitfull discussions. This work has been partially supported by the C.N.R.S. P.R.C. de Programmation, and the ESPRIT Meteor project.

7. References

[Bel 85] Belhassen S., "Séries formelles de complexité dans les types abstraits algébriques", Rapport de DEA, Orsay 1985.

[BBWT 81] Bergstra J.A., Broy M., Wirsing M., Tucker J.V., "On the power of algebraic specifications", Proc. of the M.F.C.S. Conference, L.N.C.S. 118, Springer Verlag, 1981.

[BCV 85] Bidoit M., Choppy C., Voisin F., "The Asspegique specification environment : motivations and design", Proc. 3rd Workshop on Theory and Applications of Abstract data types", Bremen, Nov. 1984. Recent Trends on Data Type Specification (H.-J. Kreowski ed.), Informatik Fachberichte 116, Springer Verlag, Berlin-Heidelberg, 1985.

[C&K 83] Choppy C., Kaplan S., "Complexity calculus for abstract data types", LRI Report n° 147, Nov. 1983.

[CLR 80] Choppy C., Lescanne P., Rémy J.-L., "Improving abstract data type specification by appropriate choice of constructors", Proc. Intern. Workshop on Program Construction, Bonas, France, 1980, Mac Millan, A. Biermann, G. Guiho and Y. Kodratoff (eds), 1983.

[Der 82] N. Dershowitz, "Orderings for term-rewriting systems", T.C.S. vol. 17.3, March 1982.

[Der 85] N. Dershowitz, "Termination", Proc. of the RTA'85 Conference, L.N.C.S. 202, 1985.

[E&M 81] Ehrig H., Mahr B., "Complexity of algebraic implementations for abstract data types", J. of Computer and System Sciences, vol 23, n° 2, Oct, 1981, pp. 223-253.

[Fla 88] Flajolet P., "Mathematical methods in the analysis of algorithms and data structures", *in* Trends in theoretical computer science, ed. Egon Borger, Computer Science Press, 1988.

[F&S 87] Flajolet P., Steyaert J.M., "A complexity calculus for classes of recursive tree algorithms", Mathematical Systems Theory, 19, 1987, pp. 301-331.

[FSZ 88] Flajolet P., Salvy B., Zimmermann P., "Lambda Upsilon Omega, an assistant algorithm analyzer", AECC-6, Rome, 1988 (proc. to appear in L.N.C.S.).

[F&G 84] Forgaard R., Guttag J.V., "REVE : a term rewriting system generator with failure-resistant Knuth-Bendix", Proc. of an NSF Workshop on the rewrite rule laboratory, Report n° 84GEN008, General Electric, Apr. 1984.

[FGJM 84] Futatsugi K., Goguen J.A., Jouannaud J.P., Meseguer J., "Principles of OBJ2", Workshop on foundations of logic and functional programming, Trento, Italy, 1986, also : CRIN Report 84-R-066.

[GSF 86] Gaudel M.C., Soria M., Froidevaux C., "Types de données et algorithmes - vol 1 : Analyse d'algorithmes, Définition des types de données", Collection Didactique, INRIA, 1986.

[GHW 85] Guttag J.V., Horning J.J., Wing J.M., "Larch in five easy pieces" Digital System Research Center Report, Jul. 1985.

[GH 86a] Guttag J.V., Horning J.J., "Report on the Larch shared language" Science of Computer Programming, vol 6, n° 2, March 1986.

[GH 86b] Guttag J.V., Horning J.J., "A Larch shared language handbook", Science of Computer Programming, vol 6, n° 2, March 1986.

[Kap 86] Kaplan S., "A compiler for conditional term rewriting systems" Proc. of the RTA'87 Conference, L.N.C.S. this volume, also : L.R.I. Report n° 315, Dec. 1986.

[Knu 68] Knuth D., "The art of computer programming : Fundamental algorithms", Addison Wesley, Reading, 1968.

[Knu 73] Knuth D., "The art of computer programming : Sorting and searching", Addison Wesley, Reading, 1973.

[M&M 78] Meir A., Moon J.W., "On the altitude of nodes in random trees", Canadian Journal of Math. 30, 1978, pp. 997-1015.

[MAPLE 85] Char B.W., Geddes K.O., Gonnet G.H., Watt S.M., "MAPLE : Reference Manual", University of Waterloo, 1985.

[S&S 84] Soria M., Steyaert J.M., "Average efficiency of pattern matching on Lisp expressions", Proc. of the CAAP 84 Conference, also : LRI Report n° 178, May 1984.

[S&F 83] Steyeart J.M., Flajolet P., "Patterns and pattern matching in trees : an analysis", Information and Control, 58, 1983.